UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/939,378	08/24/2001	Joseph Franklin Garvey	RAL920000124US1	3898

25299 7590 08/31/2004

IBM CORPORATION
PO BOX 12195
DEPT 9CCA, BLDG 002
RESEARCH TRIANGLE PARK, NC 27709

EXAMINER	
VU, TUAN A	
ART UNIT	**PAPER NUMBER**
2124	

DATE MAILED: 08/31/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/939,378	GARVEY, JOSEPH FRANKLIN	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 August 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-8* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-8* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 August 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *20010824*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed August 24, 2001.

Claims 1-8 have been submitted for examination.

Specification

2. The abstract of the disclosure is objected to because there appears to be a mistake in the expression termed as 'is utilize'. The suggested correction would be 'is utilized'. Correction is required. See MPEP § 608.01(b).

Information Disclosure Statement

3. The information disclosure statement filed 8/24/2001 fails to comply with 37 CFR 1.98(a)(2), which requires a legible copy of each U.S. and foreign patent; each publication or that portion which caused it to be listed; and all other information or that portion which caused it to be listed. As for the patent documents, such requirement has been relaxed; but as far as non-patent documents, it is still required that physical copies be provided to avert potential burden onto the examining process of the case.

In the column of listed non-patent documents in form 1449/PTO, those documents whose ID # are AB, AC, ... AG were not provided with a copy; or the so related copies are no found in the record. But Examiner has gone a step further employing special resources and means and arrived at fetching these documents online. It is henceforth urged or asked that Applicant provide the physical copies of non-patent documents for which Applicants would like to be considered.

The documents so mentioned are nevertheless considered.

Specification

4. The disclosure is objected to because of the following informalities: it appears to be a grammatical informality at "For example, in order ... structured assembly language expression, that is," (lines 5-6, pg. 12); wherein the ending ',' should be a ':".

5. The disclosure is objected to under 37 CFR 1.71, as being so incomprehensible as to almost impede a reasonable search of the prior art by the examiner. For example, the following items are not well understood: "Rule # 4 can be used to logically negate ... branch label for all tuples in expression 1, except for the last tuple is complimented ... next_and becomes next_or ... <!cc4, args4, branchto = next_or>" (pg. 11, line 35 to pg. 12, line 17). This section describes complementing labels and inverting condition codes but the way it is put together (e.g. the term 'compliment' does not indicate the ! operation expected because 'compliment' means to offer or put forth praise) does not convey the idea or enable understanding to one skill in the art, particularly based on the phraseology and sentence constructs. Examiner will base on the examples provided in terms of tuples vector being transformed to make some sense out of this deficient write-up.

Applicant is required to submit an amendment which clarifies the disclosure so that the examiner may make a proper comparison of the invention with the prior art.

Applicant should be careful not to introduce any new matter into the disclosure (i.e., matter which is not supported by the disclosure as originally filed).

Appropriate correction is required.

Claim Objections

6. Claim 1 is objected to because of the following informalities: the two 'and' at the end of lines 16, and 18 should be removed because only one 'and' (line 20) is needed to end the listing in the claim.

7. Further, claim 1 exhibits an extraneous term as in 'proceed at to the end' (line 20). The 'at' is to be removed. Appropriate correction is required.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 7 recites the limitations "said SA_Expr3's data structures" and "said SA_Expr3's last data structure" in lines 7-8, 10-11. There is insufficient antecedent basis for these limitations in the claim. This will be interpreted as if these were at best, 'data structures in SA_Expr3' and 'the last data structure in SA_Expr3', respectively.

Correction is required.

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. Claims 1, and 4-5 are rejected under 35 U.S.C. 102(b) as being anticipated by Leeper et al., "Structured Assembly Language in VAX-11 MACRO", Feb. 1986, Proceedings of the 17th SIGCSE technical symposium on Computer Science education, Vol. 18,m issue 1(hereinafter Leeper).

As per claim 1, Leeper discloses an assembler for processing structured assembly language expressions, said assembler comprising:

program code means for recognizing a structured assembly language expression's mnemonics containing elements **argl cc arg2** (e.g. *algorithm* – pg. 54; IF *X* > *Y* -pg. 54; IF *X* >= *0* pg. 55; WHILE *NAME* <> *TRAILER* – pg. 57), wherein said **cc** is a condition code (> , >=, <> -- Note: greater than, greater or equal to, not equal to are condition codes), wherein the form of said expression's mnemonics or the nature of one or more of said expression's elements selects a corresponding comparison opcode (e.g. CMPC3 – pg. 54; CMPL – pg. 55; CMPC3 – pg. 57 – Note: the form of the template expression symbols dictates a corresponding opcode), wherein said argl and said arg2 are valid arguments for said selected comparison opcode (Note: X and Y are inherently valid and type equivalent in order for opcode to function);

program code means for constructing a data structure referencing said argl, said arg2, said cc, and a branch destination (e.g. *constructs for IF-THEN, templates* – pg. 54, 3rd para);

program code means for generating a comparison opcode in response to elements of said data structure (e.g. (structured template) *CMPx* – pg. 54, 3rd para; *CMPx* – pg. 55, 1st para; *CMPx* – pg. 56, 6th para -- > (assembly language) CMPC3 – pg. 54; CMPL – pg. 55; CMPC3 – pg. 57);

program code means for generating a conditional branch based on said condition code in said data structure (e.g. *BGTR* – pg. 54, last para; *BGEQ, MNEGB* – pg. 55, 3rd para; *BNEQ, BEQL* – pg. 57, 2nd para);

program code means for generating a first branch location for execution to proceed at if said structured assembly language expression is true (e.g. *BGTR* – pg. 54, last para; *BEQ* – pg. 57, 2nd para); and

program code means for generating a second branch location for execution to proceed at if said structured assembly language expression is false (e.g. *BNEQ* - pg. 57, 2nd para); and

program code means for generating a third branch location for execution to proceed [at] to the end of said structured assembly language expression (e.g. *BEQL END_WHILE04* – pg. 57, 2nd para); and

program code means for indicating said branch destination (e.g. *WHILE04*, *END_WHILE04* – pg. 57, 2nd para) in said data structure is a branch to said first, said second, or said third branch lccations.

As per claim 4, Leeper discloses means for not generating a comparison opcode in response to said data structure (e.g. THEN_BEGINnn: NOP ... ELSE_BEGINnn: ... pg. 55, top para)

As per claim 5, Leeper discloses assembling code generation by iterating over a vector of structured assembly language (SAL) structures of various forms (e.g. FOR LOOP CONSTRUCT, the WHILE LOOP CONSTRUCT, REPEAT-UNTIL CONSTRUCT – pg. 55-58 – Note: the combination of more than one logical expressions to make compounded logical expressions is implicitly disclosed in every programming language with complex iteration and

compounded arguments type logical operations, hence building assembly language from combination of vectors like SAL templates as taught by Leeper is disclosed via putting together complex iteration statements and comparing compounded expressions).

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 2-3, and 6-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Leeper et al., "Structured Assembly Language in VAX-11 MACRO", Feb. 1986, Proceedings of the 17th SIGCSE technical symposium on Computer Science education, Vol. 18,m issue 1,pp. 53-60; in view of Curzon, Paul, "A Verified Compiler for a Structured Assembly Language", 1992, *International Workshop on Higher Order Logic Theorem Proving and Its Applications*, pp. 253-269(hereinafter Curzon) .

As per claim 2, Leeper discloses assembler further includes program code means for recognizing a structured assembly language expression's mnemonics having a format of condition code (re claim 1; $X > Y$ -pg. 54; IF $X >= 0$ pg. 55) but does not explicitly disclose a **cc** form, wherein said cc is a condition code. Abstracting a operator into a more symbolic form ((e.g. <var1> <op> <var2>) was a known concept in high-level language at the time the invention was made and structured assembly language as taught by Leeper is such a form of high-level language with respect to assembly code. Hence, abstracting the condition operator or symbol as suggested by Leeper into a more generic CC form is further enhanced by Curzon (e.g.

TransWhile bcode ccode base size - pg. 259, top L para) who also teaches compiler for processing and using a Structured Assembly Language (pg. 258-260). It would have been obvious for one of ordinary skill in the art at the time the invention was made to generate the template as taught by Leeper so that instead of using condition symbol, a cc form as suggested by Curzon represents such condition code because this way the more generic **cc** form can further be translated into a wider range of conditional situations requiring more elaborate operators or symbolology such as taught by known practices of high-language abstraction, thus expanding the useability of condition codes that would otherwise be more limited in Leeper's approach.

As per claim 3, Leeper does not disclose explicitly generating a data structure referencing no arguments, cc, and a branch destination in response to the condition code. But the CC limitation is taught from the teaching of Curzon; hence this teaching would have been obvious in view of the rationale as set forth in claim 2. Leeper, however discloses a condition code without arguments and a destination for branch (see *BRW ELSEBEGINnn* - top para pg. 55 – Note: a 'branch always' is also a condition code wherein no arguments are needed because it is like a if (TRUE) type of assertion); hence this data structure referencing no arguments, a CC, and a destination would also have been obvious by virtue of Leeper's teachings combined with the rationale using Curzon.

As per claim 6, Leeper discloses means:

for recognizing a structured assembly language expression's mnemonics resulting from a logical ANDing of SA_Expr1 and SA_Expr2, wherein each of said SA_Expr1 and said SA_Expr2 is a unit or compound structured assembly language expression (Note: the use of AND operator for logical AND is implicitly disclosed because all high-level languages leading

to an assembly language have this operator (e.g. '&&' or AND or inverse-V-Notation) operating on at least 2 arguments to make a single unit of SAL as represented in Leeper's templates);

for setting said branch in each data structure of said SA_Expr1 that is branching to said first branch location to branch to end of said SA_Expr1 (e.g. *THEN_BEGINnn:* – pg. 54, 3rd para; *THEN_BEGIN01:* – pg. 54, last para – Note: from the standpoint in high-level code parsing, branching by taking all the contents of the node to another node, like skipping the entirety of the contents of a subtree in flow graph, i.e. the unexecuted instructions above the adjusted branch destination label, implicitly discloses this limitation).

The limitation for concatenating and preserving order of data structures in said SA_Expr1 and said SA_Expr2 into a single compound structured assembly language expression falls under the known concept of programming languages which teaches a compound logical operation is such that it concatenates orderly simpler logical operations as has been addressed in claim 5; and should be inferred to be disclosed by Leeper; however, this is not explicitly shown by Leeper.

In case Leeper does not teach a compound SAL expression as inferred from high-level programming language, this limitation is taught by Curzon (see pg. 260 R column, 261, R column – Note: code using inverse V for AND operators stands for ANDing in structured language compound expressions). Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide such Logical compound expression thus taught to the template generating by Leeper because this would enhance the logical AND operation so that a number of simple condition can be addressed separately to yield a result by virtue of a compound condition checking as well-known in the art of programming language.

14. Claims 7-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Leeper et al., "Structured Assembly Language in VAX-11 MACRO", Feb. 1986, Proceedings of the 17th SIGCSE technical symposium on Computer Science education, Vol. 18,m issue 1; and Curzon, Paul, "A Verified Compiler for a Structured Assembly Language", 1992, *International Workshop on Higher Order Logic Theorem Proving and Its Applications*, pp. 253-269; and further in view of Mangelsdorf, USPN: 6,012,836 (hereinafter Mangelsdorf).

As per claim 7, Leeper discloses means for

recognizing a structured assembly language expression's mnemonics requiring a logical ORing of SA_Expr3 and SA_Expr4,wherein each of said SA Expr3 and said SA Expr4 is a unit structured assembly language expression (Note: the use of OR operator for logical OR is implicitly disclosed because all high-level languages to be translated into assembly code have this operator (e.g. '||' or OR or V-Notation) operating on at least 2 arguments to make a single unit of SAL as represented in Leeper's templates); and

concatenating and preserving order of data structures in said SA Expr3 and said SA Expr4 into a single compound structured assembly language expression (Note: this limitation would have been implicit or at worst obvious in view of the rationale using Curzon applied to the AND operation from above).

Since Leeper does not explicitly teach compound SAL expression when applying the OR-ing operation thereto, this limitation would also have been obvious in view of the rationale to use the teachings by Curzon (e.g. pg. 260 R column, 261, R column – with the *V* operator standing for OR-ing).

Leeper does not explicitly disclose:

(i) changing said branch location in each of the data structures of SA_Expr3, except the last data structure of SA_Expr3, from said branch location to end of SA_Expr3

(ii) complementing said branch condition in said SA Expr3's last data structure

(iii) changing said branch location in said SA Expr3's last data structure from a branch to said first location to branch to said second location, or from a branch to said second location to branch to said first location.

But based on the understanding from the specifications, these limitations evolve around changing the unit SAL expressions belonging to a compound expression by complementing the condition code operator for each unit SAL expressions and swapping the destination address. And this is reminiscent of an overall expression comprised of '&&' and '|| ' operations wherein modifications of the operators are compensated with the inverting of the partial global result, such result in this instance being represented by a destination address based a TRUE or FALSE state (or partial global result) of the combined sub-expressions evaluation from the && and || operations. And by inverting the outcome while complementing the internal operators, the effect is the same as applying a variance of the DeMorgan's theorem, a well-known concept at the time the invention was made. Official notice is taken that the use of DeMorgan's theorem enabling swapping of logical operators to accommodate for environment with restraint in the hardware implementation of specific operator was a known concept at the time the invention was made. For instance, Mangelsdorf, in a method to accommodate for hardware deficiencies, teaches using DeMorgan's approach to eliminate of the NOT operations (col. 16, lines 19-37). In view of the benefits imparted to complementing operators in the inside logical operations and inverting the outside state as known to DeMorgan's theorem and Mangelsdorf's approach, these limitations,

i.e. (i) , (ii) and (iii) would have been obvious because applying a variance of the above theorem such as to complementing and inverting the internal logical operators and the outside state, respectively, would yield hardware related benefits favoring a certain instruction set or architecture according to the above Official notice or Mangelsdorf, thus enhancing the assembly language generation for a particular platform machine in which code generation and resources have to be optimized in view of the constraints above.

As per claim 8, this claim corresponds to the limitations of claim 7 for it also include changing/swapping destination location and complementing branch condition code and further includes the use of complementing to substitute for a would-be NOT logical operation as suggested by Mangelsdorf; hence is rejected using the rationale applied to claim 7.

Conclusion

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

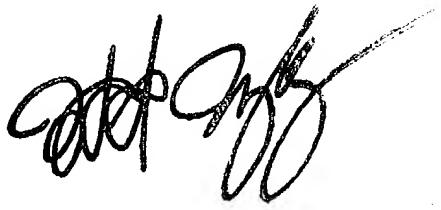
(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please consult Examiner before using this number)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
August 17, 2004



TODD INGBERG
PRIMARY EXAMINER